

What Is Claimed Is:

1. A method for configuring computer program including at least one functional unit, characterized by the following steps:
  - creation of at least one implementation-independent configuration data file (1) and/or alteration of information filed in the at least one implementation-independent configuration data file (1);
  - automatic set-up and/or automatic update of configuration data, stored in a configuration data container (3), as a function of the information filed in the at least one implementation-independent configuration data file (1);
  - automatic generation of at least one implementation-dependent configuration data file (5) as a function of the configuration data stored in the configuration data container (3);
  - automatic configuration of the at least one functional unit as a function of information filed in the at least one implementation-dependent configuration data file.
2. The method as recited in Claim 1, wherein at least one item of dependency information, which describes a dependency on at least two configuration data present in the configuration data container, is generated automatically, and the at least one implementation-dependent configuration data file is generated as a function of the at least one item of dependency information.
3. The method as recited in one of the preceding claims, the computer program having a plurality of functional units, wherein a plurality of implementation-independent configuration data files is created, and each of the implementation-independent configuration data files is assigned to at least one functional unit.
4. The method as recited in one of the preceding claims, the computer program having a plurality of functional units, wherein a plurality of implementation-dependent configuration data files is generated,

and each of the implementation-dependent configuration data files is assigned to at least one functional unit.

5. The method as recited in one of the preceding claims,  
wherein the at least one implementation-dependent configuration data file is generated as a function of at least one property of hardware on which an installation of at least a portion of the configured computer program is to be made possible.
6. The method as recited in one of the preceding claims,  
wherein the at least one implementation-dependent configuration data file is generated as a function of the result of a plausibility check.
7. The method as recited in Claims 5 and 6,  
wherein the at least one hardware property is used for carrying out the plausibility check.
8. The method as recited in one of the preceding claims,  
wherein a documentation is created automatically, and the documentation describes the information filed within the at least one implementation-independent configuration data file and/or the at least one implementation-dependent configuration data file.
9. The method as recited in one of the preceding claims,  
wherein the at least one implementation-independent configuration data file is created in an XML-based format.
10. The method as recited in one of the preceding claims,  
wherein as a function of the configuration data, it is automatically determined whether a functional unit included by the computer program is needed by the computer program, and this functional unit is only configured if the functional unit is needed by the computer program.
11. A software system for configuring a computer program including at least one functional unit,  
wherein the software system comprises:
  - at least one implementation-independent configuration data file;

- a configuration data container including configuration data, and/or means for creating a configuration data container as a function of information filed in the at least one implementation-independent configuration data file;
  - means for altering and/or reading out configuration data from the configuration data container;
  - means for automatically generating at least one implementation-dependent configuration data file as a function of configuration data stored in the configuration data container; and
  - means for automatically configuring the at least one functional unit as a function of information filed in the implementation-dependent configuration data file.
12. The software system as recited in Claim 11, wherein the software system has means for carrying out a method as recited in one of Claims 2 through 11.
  13. The software system as recited in Claim 11 or 12, wherein the software system is stored in a storage medium.
  14. The software system as recited in Claim 13, wherein the software system is stored in a random access memory, in a read-only memory or in a flash memory.
  15. The software system as recited in Claim 13, wherein the software system is stored on a digital versatile disk (DVD), a compact disk (CD) or on a hard disk.
  16. A computing element, in particular a control device, having a microprocessor, wherein the computing element is programmed for carrying out a method as recited in one of Claims 1 through 10.